

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE (USPTO)	
Application Serial Number	10/715,804
Confirmation Number	5203
Filing Date	November 18, 2003
Title of Application	Application Model that Integrates the Web Experience with the Traditional Client Application Experience
First Named Inventor	Margaret L. Goodwin
Assignee	Microsoft Corporation
Group Art Unit	2193
Examiner	Jason D. Mitchell
Attorney Docket Number	MS1-1791US

To: **Mail Stop: APPEAL BRIEFS – PATENTS**

Commissioner for Patents
 PO Box 1450
 Alexandria, VA 22313-1450

From: Colin D. Barnitz (Tel. 512-505-8162 x5002; Fax 509-323-8979)
 Lee & Hayes, PLLC
 601 W. Riverside Ave, Suite 1400
 Spokane, WA 99201

Customer Number 22801

APPEAL BRIEF

This appeal is taken from the rejection of pending claims 1, 4, and 6-13 set forth in the Office Action dated February 1, 2010.

TABLE OF CONTENTS

<u>Headings</u>	<u>Page No.</u>
I. Real Party in Interest.....	3
II. Related Appeals and Interferences.....	3
III. Status of Claims.....	3
IV. Status of Amendments.....	3
V. Summary of the Claimed Subject Matter.....	4
VI. Grounds of Rejection to be Reviewed on Appeal.....	6
VII. Argument.....	6
VIII. Claims Appendix.....	19
IX. Evidence Appendix.....	22
X. Related Proceedings Appendix.....	23
XI. Conclusion.....	18
XII. Fees.....	18

I. REAL PARTY IN INTEREST

The real party in interest in this application is the assignee of record, Microsoft Corporation of Redmond, Washington, USA.

II. RELATED APPEALS AND INTERFERENCES

Appellant and Appellant's representative are not aware of any related prior or pending appeals, judicial proceedings or interferences which may be related to, directly affect, be directly affected by, or may have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1, 4, and 6-13 are currently pending. Claims 2, 3, and 5 have been canceled. Claims 14-23 were previously withdrawn. All of pending Claims 1, 4, and 6-13 stand rejected and have been at least twice rejected. Accordingly, the rejection of Claims 1, 4, and 6-13 is being appealed.

IV. STATUS OF AMENDMENTS

No amendments have been filed subsequent to the rejection of the pending claims by the Office Action mailed February 1, 2010.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

A concise explanation of the subject matter defined in each independent claim is provided below by referring to the specification by paragraph number and, where applicable, to the drawings by reference character, to guide the Board in its understanding of the claimed subject matter. However, Appellant respectfully notes that additional support for the claims is provided throughout the specification and drawings, and is not limited to that provided in this concise summary.

A. Summary of the Subject Matter of Independent Claim 1

Independent Claim 1 is directed to a computer-readable storage medium (e.g., FIG. 5, items 504, 509, 510; page 32, lines 19-23; and page 33, lines 2-17) having stored thereon an application framework (e.g., FIG. 1, item 103; page 4, lines 14-24) for developing an application (e.g., FIG. 3, item 301; page 13, lines 20-22), comprising: an application object (e.g., page 14, lines 10-13) that isolates the application from other applications or external resources (e.g., page 14, lines 14-16), raises startup and shutdown events for the application (e.g., page 14, lines 18-19), and manages application windows and resources (e.g., page 14, lines 17-18); navigation components (e.g., FIG 3, items 315, 321, 325; page 19, line 24; page 22, line 22; page 24, line 23) that provide navigation functionality by sharing a global state across a plurality of pages (e.g., page 20, lines 4-6), journaling (e.g., page 25, lines 3-4), journal extensibility (e.g., page 5, line 3), and structured navigation (e.g., page 26, lines 20-22); application lifecycle management components (e.g., FIG. 1, item 101; page 5, lines 8-11) that define how the application is deployed, installed, activated, updated, rolled back, and

removed from a computing system (e.g., page 5, lines 8-11); a secure execution environment (e.g., FIG. 4, item 420; page 30, line 18) that defines a default set of permissions (e.g., page 30, lines 18-20) for the application during execution of the application in the secure execution environment, and if the application requires permissions in addition to the default set of permissions, requiring installation of the application (e.g., page 31, lines 6-7); a component (e.g., FIG. 3, item 307; page 18, lines 10-12) that defines a mechanism that allows the application to access common window properties of a hosting environment in a like manner regardless of whether the hosting environment is a browser or a standalone window environment; and a manifest (e.g., FIG. 2, item 211; page 5, lines 23-25) that specifies a first subset of components of the application as required, a second subset of components of the application as on-demand, and a third subset of components of the application as online(e.g., page 9, lines 2-3), with the first, second, and third subsets of components of the application differing (e.g., page 9, lines 2-3), wherein the second subset of components being drizzle-downloaded in the background as a user interacts with the application (e.g., page 9, lines 13-14), wherein when a specific component of the second subset of components is requested, the specific component takes precedence over remaining components of the second subset of components and is downloaded on-demand while the remaining components are drizzle-downloaded in the background (e.g., page 9, lines 14-17).

VI. GROUNDINGS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to be reviewed on appeal are:

Whether Claims 1, 4 and 6-12 are obvious under 35 U.S.C. § 103(a) over Kraenzel et al. (U.S. Patent No. 6,742,026 – hereafter “Kraenzel”) in view of Li et al. (U.S. Patent Application Publication No. 2003/0101445 – hereafter “Li”).

Whether Claim 13 is obvious under 35 U.S.C. § 103(a) over Kraenzel in view of Li and Gamo (U.S. Patent Application Publication No. 2004/0107291 – hereafter “Gamo”).

VII. ARGUMENT

REJECTION OF CLAIMS 1, 4 and 6-12 UNDER 35 U.S.C. §103(a)

A. Rejection of Independent Claim 1 under 35 U.S.C. §103(a)

Independent Claim 1 stands rejected as being unpatentable over Kraenzel in view of Li. The rejection of Independent Claim 1 should be reversed at least because the Office has failed to establish a *prima facie* case of unpatentability of Independent Claim 1. As will be shown below, the Office has ignored express recitations of Independent Claim 1, and the Office has provided mere conclusory statements to attempt to meet the recitations of Independent Claim 1.

Appellant’s Independent Claim 1 reads as follows (with emphasis added):

1. A computer-readable storage medium having stored thereon an application framework for developing an application, comprising:

an application object that isolates the application from other applications or external resources, raises startup and shutdown

events for the application, and manages application windows and resources;

navigation components that provide navigation functionality by sharing a global state across a plurality of pages, journaling, journal extensibility, and structured navigation;

application lifecycle management components that define how the application is deployed, installed, activated, updated, rolled back, and removed from a computing system;

a secure execution environment that defines a default set of permissions for the application during execution of the application in the secure execution environment, and if the application requires permissions in addition to the default set of permissions, requiring installation of the application;

a component that defines a mechanism that allows the application to access common window properties of a hosting environment in a like manner regardless of whether the hosting environment is a browser or a standalone window environment; and

a manifest that specifies a first subset of components of the application as required, a second subset of components of the application as on-demand, and a third subset of components of the application as online, with the first, second, and third subsets of components of the application differing,

wherein the second subset of components being drizzle-downloaded in the background as a user interacts with the application, wherein when a specific component of the second subset of components is requested, the specific component takes precedence over remaining components of the second subset of components and is downloaded on-demand while the remaining components are drizzle-downloaded in the background.

1. The combination of Kraenzel with Li fails to teach or suggest “a secure execution environment that defines a default set of permissions for the application during execution of the application in the secure execution environment, and if the application requires permissions in addition to the default set of permissions, requiring installation of the application”

In the Office Action mailed February 1, 2010 (hereinafter “the Office Action”), the Office has failed to point to any portion of the cited documents that teaches or suggests at least “a secure execution environment that defines a default set of permissions for the application during execution of the application in the secure execution environment, and if the application requires permissions in addition to the default set of permissions, requiring installation of the application,” as recited in Appellant’s Claim 1. For example, with in rejecting the above recitation of Appellant’s Claim 1, the Office states the following:

a secure execution environment that defines a default set of permissions for the application during execution of the application in the secure execution environment, and if the application requires permissions in addition to the default set of permissions, requiring installation of the application (column 39, lines 15-32, offline subscription denoting additional permissions and secure environment as well); and

Appellant respectfully notes that col. 39, lines 15-32, of Kraenzel merely describes the following (emphasis added):

The end user opens a Domino Web application that the Web site developer and administrator have enabled for offline use. In the initial

screen of the Web application, the user clicks a Web control that contains the words, "Go Offline." A pop-up menu appears giving the user the option of installing the Web application as a subscription on the local machine (client). A subscription includes the Domino Web application, its related databases, and subscription property settings.

If this is the first time an end user is installing a subscription, the Lotus iNotes Sync Manager utility is downloaded to the user's machine. ***Files needed for working offline in a secure environment and for managing synchronization are also downloaded.*** The installation is seamless and nearly invisible to the end user. If download time is an issue, the end user can also use a CD to install Lotus iNotes Sync Manager. If the end user already has a Notes ID, that ID can be used; if not, a new ID can be generated.

From a review of the cited portions of Kraenzel, and the remainder of Kraenzel, Appellant submits that Kraenzel does not teach or suggest "a secure execution environment ***that defines a default set of permissions*** for the application during execution of the application in the secure execution environment, ***and if the application requires permissions in addition to the default set of permissions, requiring installation of the application,***" as recited in Appellant's Claim 1. Instead, Kraenzel merely describes that "[f]iles needed for working offline in a secure environment and for managing synchronization are also downloaded" (Kraenzel, col. 39, lines 26-28). Consequently, there is no teaching or suggestion of defining a default set of permissions during execution in the secure execution environment, and "if the application requires permissions in addition to the default set of permissions, requiring installation of the application." For example, merely downloading files for working offline, as described by Kraenzel, does not teach or suggest "a default set of

permissions for...the secure execution environment,” or that “if the application requires permissions in addition to the default set of permissions, requiring installation of the application,” as recited in Appellant’s Claim 1. Accordingly, Kraenzel fails to teach or suggest at least the above recitation of Appellant’s Claim 1. Li merely describes subdividing programs into autonomous modules (e.g., Li, Abstract), and Li fails to compensate for the shortcomings in Kraenzel pointed out above. Thus, Claim 1 is allowable over the cited documents for at least this recitation.

2. The combination of Kraenzel with Li fails to teach or suggest “wherein the second subset of components being drizzle-downloaded in the background as a user interacts with the application, wherein when a specific component of the second subset of components is requested, the specific component takes precedence over remaining components of the second subset of components and is downloaded on-demand while the remaining components are drizzle-downloaded in the background”

The Office has failed to point to any portion of the cited documents that teaches or suggests at least “the second subset of components being drizzle-downloaded in the background as a user interacts with the application, wherein when a specific component of the second subset of components is requested, the specific component takes precedence over remaining components of the second subset of components and is downloaded on-demand while the remaining components are

drizzle-downloaded in the background,” as recited in Appellant’s Claim 1. For example, at page 6, lines 1-2 of the Office Action, the Office acknowledges that Kraenzel does not disclose a first “subset of components as required and a second subset of components as on-demand.” Instead, the Office asserts at page 6, lines 4-19, of the Office Action that the above recitation of Appellant’s Claim 1 is taught by Li at paragraphs 0046, 0048, 0053, 0060, and FIG. 10. However, Appellant notes that the portions of Li cited by the Office as allegedly teaching the above recitation of Claim 1 actually read as follows:

[0046] Moving to FIG. 4, block diagram 158 illustrates the list of modules downloaded in operation 146 of FIG. 3 in accordance with one embodiment of the invention. It should be understood that the list of modules illustrated in block diagram 158 are simply illustrative of a number of modules, and thus, any number of modules may exist depending on the software application and use thereof. With the foregoing in mind, application 160, core modules and multiple non-core modules are shown. Here, it is shown that core module version 1 162 has been superseded by core module version 2 164. Accordingly, the list of modules downloaded in operation 146 of FIG. 3 includes only the most recent version of the core module, i.e. version 2 164 in one embodiment. It should be appreciated that core module version 1 is being displayed for illustrative purposes only. The list of FIG. 4 further includes modules 1-5, represented as blocks 166-174. In one embodiment of the invention, the list of modules used by the application is compared to the modules on the local system to determine which modules the system requires to be downloaded.

[0048] FIG. 5 illustrates flowchart 176 displaying a method where modules are downloaded and installed when needed as the application is running in accordance with one embodiment of the invention. Flowchart 176 initiates with operation 178 where the application is running on the local system. It should be appreciated that as the user is running the application and utilizing different functionality, eventually a module besides the core module will be required. Following the example of a printer, a class not included in the core module may be required for some operation such as printing, editing, formatting, etc. The method then advances to operation 180 where the module containing the required class is identified. The

method moves to operation 182 where the module containing the required class is downloaded and installed on the local system.

[0053] The above illustrative example is also applicable to other non-core modules which allow for feature functionality. Once the classes comprising the core module are installed a user may elect to perform tasks which require feature functionality. In the example of a printer and its applications, the user may wish to rotate or print a graphic. Once the rotate or print button is activated, then code will ask the Java Virtual Machine for a class enabling the feature functionality. Accordingly, the Java Virtual Machine will request the class required for the feature functionality and operations 188-196 will be repeated for the required class and all interrelated classes. As will be explained in further detail in reference to FIGS. 8-10, the modules enabling feature functionality may also be downloaded in the background prior to being requested by the user in another embodiment.

[0060] Alternatively, if the module requested is not the module being downloaded by the download manager in decision operation 234 of FIG. 10, then the method proceeds to operation 240 where the current download is suspended. For example, a user wishes to rotate an image and needs module 4 (M_4) for the feature functionality. However, the module being downloaded by the download manager is module 3 (M_3). The download of M_3 is suspended in operation 240 so that M_4 may be downloaded without competing with M_3 for bandwidth. In one embodiment of the invention, the download manager logic comprising this decision operation is only initiated when the user interrupts the methodical download of the modules as governed by the priority list. The method advances to operation 242 where the requested module is downloaded. Continuing with the example above, M_4 would be downloaded here as M_3 is suspended. In one embodiment, another instantiation of the code which was downloading M_3 is used to download M_4 . In operation 244, the requested module is installed. The method moves to operation 246 where the thread for the suspended download is resumed. In reference to the above example, the downloading of M_3 is resumed upon the installation of M_4 . Then in operation 248, the method proceeds to operation 220 of FIG. 8 where the module is downloaded.

Additionally, FIG. 10 of Li is reproduced below for convenience.

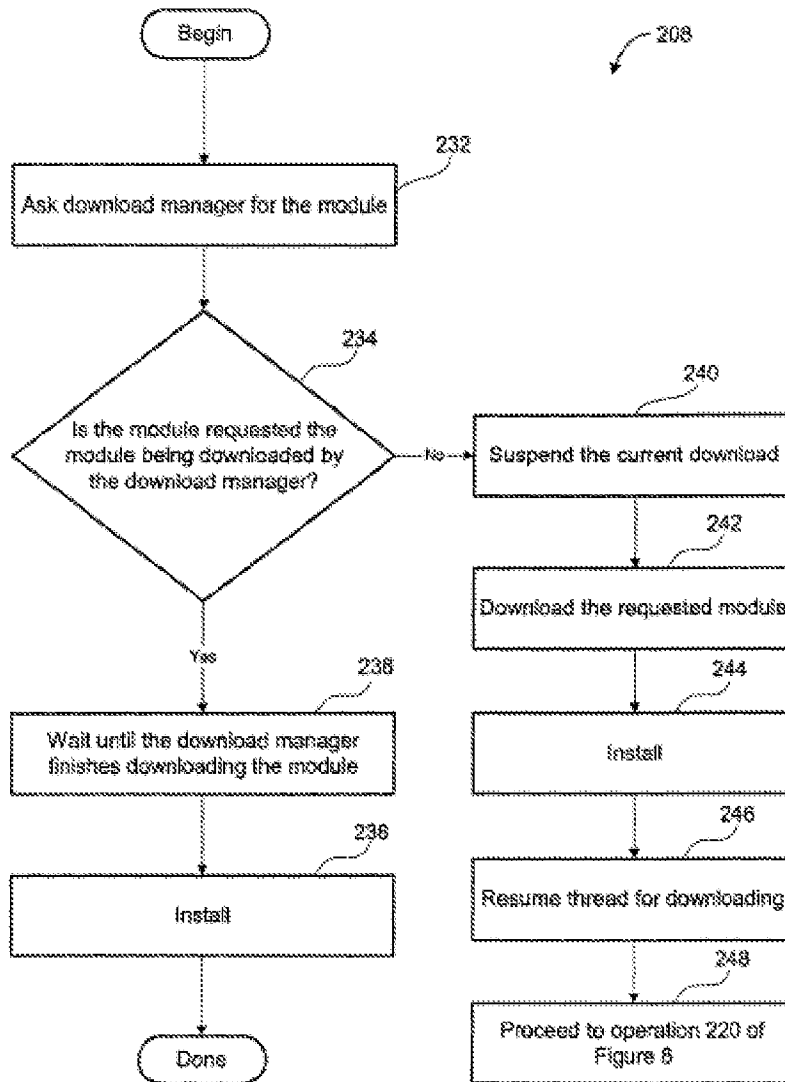


FIG. 10

The cited portion of Li describes that “if the module requested is not the module being downloaded...the current download is suspended” (par. 0060). More specifically, the downloading of a module M_3 is **suspended** such that a module M_4 may be downloaded without competing with M_3 for bandwidth (par. 0060). On the other hand, Appellant’s Claim 1 recites “the specific component takes precedence over remaining components...and is downloaded on-demand **while the remaining**

components are drizzle-downloaded in the background” (emphasis added). Li does not describe the module M₄ being downloaded on demand while module M₃ is downloaded in the background. Further, Li teaches away from module M₄ being downloaded on demand while module M₃ is downloaded in the background by advocating **suspending** downloading of module M₃.

Moreover, Li describes “the modules enabling feature functionality may also be downloaded in the background prior to being requested by the user in another embodiment” (par. 0053). However, Li does not describe the modules as being on-demand modules, much less a specific module taking priority over remaining modules when the specific module is requested, and downloaded on-demand while the remaining modules are drizzle-downloaded in the background. Li merely describes having modules downloaded prior to requesting them. Downloading modules prior to being requested is not the same as “the specific component takes precedence over remaining components of the second subset of components and is downloaded on-demand while the remaining components are drizzle-downloaded in the background,” as Claim 1 recites. Accordingly, Li does not teach or suggest “the second subset of components being drizzle-downloaded in the background as a user interacts with the application, wherein when a specific component of the second subset of components is requested, the specific component takes precedence over remaining components of the second subset of components and is downloaded on-demand while the remaining components are drizzle-downloaded in the background,” as Claim 1 recites.

Moreover, Kraenzel fails to compensate for the shortcomings in Li pointed out above. For example, Kraenzel merely describes “Lotus iNotes Sync Manager is

integrated into the Windows desktop and lets the user manage multiple offline subscriptions” (Col. 39, lines 34-36). Kraenzel further describes “[s]ynchronizing the online and offline versions of the application with each other” (Col. 39, lines 42-43). However, Kraenzel makes no mention of the application having three subsets of components of the application categorized as required, on-demand, and online, much less that the on-demand components are drizzle-downloaded in the background as the user interacts with the application. Thus, Kraenzel also does not teach or suggest “the second subset of components being drizzle-downloaded in the background as a user interacts with the application, wherein when a specific component of the second subset of components is requested, the specific component takes precedence over remaining components of the second subset of components and is downloaded on-demand while the remaining components are drizzle-downloaded in the background,” as recited in Claim 1.

In view of the foregoing, Appellant respectfully asserts that Claim 1 is allowable over the combination of Kraenzel and Li.

B. Rejection of Dependent Claims 4, 6-9, 11 and 12 under 35 U.S.C. §103(a)

Dependent Claims 4, 6-9, 11 and 12 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kraenzel in view of Li. Claims 4, 6-9, 11 and 12 depend from Independent Claim 1. As discussed above, Claim 1 is allowable over the combination of Kraenzel and Li. Therefore, dependent Claims 4, 6-9, 11 and 12 are patentable at least because they depend from an allowable base claim.

C. Rejection of Dependent Claim 10 under 35 U.S.C. §103(a)

Dependent Claim 10 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Kraenzel in view of Li. Claim 10 includes “wherein the component that provides journaling and journal extensibility comprises a Journal object.” Further, Claim 10 depends from Claim 1, which recites “navigation components that provide...journaling.” Consequently, Appellant’s journaling is provided by a Journal object that is a navigation component. Thus, the Journal object provides journaling as a navigation component, i.e., navigation journaling. The Office asserts that Kraenzel teaches Claim 10, citing col. 5, line 35, of Kraenzel. However, the cited portion of Kraenzel merely discusses the following:

Domino online services (DOLS) 62 is used by a web site administrator to configure Internet Notes (iNotes) clients to auto download from server 60, thus providing iNotes clients with web access using HTTP with various browsers, and with local processing and replication.

From a review of the cited portion of Kraenzel, and the remainder of Kraenzel, Appellant respectfully submits that the Office has failed to cite any portion of Kraenzel that teaches or suggests a navigation component that provides journaling, “wherein the component that provides journaling and journal extensibility comprises a Journal object,” as recited in Appellant’s Claim 10. Instead, Kraenzel merely describes that Internet Notes clients are configured to auto download from a server (col. 5, line 35). Thus, there is no teaching or suggestion of a Journal object that is a navigation component that provides journaling and journal extensibility. Consequently, Appellant respectfully submits that Claim 10 is additionally allowable over the combination of Kraenzel and Li for at least this reason.

REJECTION OF CLAIM 13 UNDER 35 U.S.C. §103(A)

A. Rejection of Dependent Claim 13 under 35 U.S.C. §103(a)

Dependent Claim 13 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Kraenzel in view of Li and Gamo. Claim 13 depends from Independent Claim 1. As discussed above, Claim 1 is allowable over Kraenzel in view of Li. Further, Gamo is cited for describing that “the cache can be used, and thus the download from the server can be partly omitted when the program is executed at the next time” (Gamo, par. 0068). However, Gamo does not cure the deficiencies of Kraenzel and Li noted above with respect to Claim 1. Therefore, dependent Claim 13 is allowable over Kraenzel in view of Li and Gamo at least due to its dependence on allowable Claim 1.

XI. CONCLUSION

In view of the foregoing Arguments, Appellant respectfully requests reversal of the rejections of Claims 1, 4, and 6-13, and issuance of a timely Notice of Allowance.

XII. FEES

Fees will be paid by credit card through the EFS Web; however, Appellant hereby authorizes the Commissioner to charge any deficiency of fees and credit any overpayments, including any fees for extensions of time, to Deposit Account Number 12-0769.

Respectfully Submitted,

Lee & Hayes, PLLC
Representative for Appellant

_____/Colin D. Barnitz 35061/
Colin D. Barnitz
(colin@leehayes.com; 512-505-8162 x5002)
Registration No. 35061

Dated: _____ September 14, 2010 _____

VIII. CLAIMS APPENDIX

1. A computer-readable storage medium having stored thereon an application framework for developing an application, comprising:

an application object that isolates the application from other applications or external resources, raises startup and shutdown events for the application, and manages application windows and resources;

navigation components that provide navigation functionality by sharing a global state across a plurality of pages, journaling, journal extensibility, and structured navigation;

application lifecycle management components that define how the application is deployed, installed, activated, updated, rolled back, and removed from a computing system;

a secure execution environment that defines a default set of permissions for the application during execution of the application in the secure execution environment, and if the application requires permissions in addition to the default set of permissions, requiring installation of the application;

a component that defines a mechanism that allows the application to access common window properties of a hosting environment in a like manner regardless of whether the hosting environment is a browser or a standalone window environment; and

a manifest that specifies a first subset of components of the application as required, a second subset of components of the application as on-demand, and a third subset of components of the application as online, with the first, second, and third

subsets of components of the application differing, wherein the second subset of components being drizzle-downloaded in the background as a user interacts with the application, wherein when a specific component of the second subset of components is requested, the specific component takes precedence over remaining components of the second subset of components and is downloaded on-demand while the remaining components are drizzle-downloaded in the background.

4. A computer-readable storage medium as recited in claim 1, wherein the application framework further includes components that define the behavior of windows associated with the application.

6. A computer-readable storage medium as recited in claim 1, wherein the component that provides navigation functionality comprises a NavigationApplication object.

7. A computer-readable storage medium as recited in claim 6, wherein the NavigationApplication object identifies an initial resource to which the application navigates when launched.

8. A computer-readable storage medium as recited in claim 7, wherein the NavigationApplication object further includes navigation related events that are fired in response to the occurrence of a navigation.

9. A computer-readable storage medium as recited in claim 7, wherein the NavigationApplication object further comprises a Properties collection in which is stored state information about the application.

10. A computer-readable storage medium as recited in claim 1, wherein the component that provides journaling and journal extensibility comprises a Journal object.

11. A computer-readable storage medium as recited in claim 1, wherein the navigation framework further comprises a NavigationWindow component associated with the application and that persists across navigations.

12. A computer-readable storage medium as recited in claim 1, wherein the first subset of components are minimum code for the application to run in the hosting environment.

13. A computer-readable storage medium as recited in claim 1, wherein the third subset of components are stored in a transient cache.

IX. EVIDENCE APPENDIX

None

X. **RELATED PROCEEDINGS APPENDIX**

None